



```
IIIIII  NN  NN  PPPPPPP  UU  UU  TTTTTTTTTT
IIIIII  NN  NN  PPPPPPP  UU  UU  TTTTTTTTTT
II      NN  NN  PP      PP  UU  UU  TT
II      NN  NN  PP      PP  UU  UU  TT
II      NNNN  NN  PP      PP  UU  UU  TT
II      NNNN  NN  PP      PP  UU  UU  TT
II      NN  NN  PPPPPPP  UU  UU  TT
II      NN  NN  PPPPPPP  UU  UU  TT
II      NN  NN  PP      PP  UU  UU  TT
II      NN  NN  PP      PP  UU  UU  TT
II      NN  NN  PP      PP  UU  UU  TT
II      NN  NN  PP      PP  UU  UU  TT
IIIIII  NN  NN  PP      PP  UU  UU  TT
IIIIII  NN  NN  PP      PP  UU  UU  TT
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  !IIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

.....

```
0001 0 %title 'INPUT - Analyze and Act on Input'
0002 0      module input (
0003 1          ident='V04-000') = begin
0004 1
0005 1
0006 1 *****
0007 1 *
0008 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 *  ALL RIGHTS RESERVED.
0011 1 *
0012 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 *  TRANSFERRED.
0018 1 *
0019 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 *  CORPORATION.
0022 1 *
0023 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *
0027 1 *****
0028 1
0029 1
0030 1 **
0031 1 Facility:    VAX/VMS Telephone Facility, Analyze and Act on Input
0032 1
0033 1 Abstract:    This module receives all of the terminal input from
0034 1              the user and decides what to do with it.  It handles
0035 1              both conversation text and commands.
0036 1
0037 1
0038 1 Environment:
0039 1
0040 1 Author: Paul C. Anagnostopoulos, Creation Date: 10 November 1980
0041 1
0042 1 Modified By:
0043 1
0044 1          V03-001 PCA1020      Paul C. Anagnostopoulos 24-May-1983
0045 1          Support 8-bit characters.
0046 1          Add PHONE as a synonym for the DIAL command.
0047 1 --
```



INPUT  
V04-000

INPUT - Analyze and Act on Input  
Module Declarations

K 1  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 2  
(2)

```

49      0048 1 %sbttl 'Module Declarations'
50      0049 1
51      0050 1   Libraries and Requires:
52      0051 1
53      0052 1
54      0053 1   library 'sys$library:starlet.l32';
55      0054 1   library 'sys$library:tpamac.l32';
56      0055 1   require 'phonereq';
57      0384 1
58      0385 1
59      0386 1   Table of Contents:
60      0387 1
61      0388 1
62      0389 1   forward routine
63      0390 1       phn$kbd_route: novalue,
64      0391 1       phn$cmd_parse: novalue;
65      0392 1
66      0393 1
67      0394 1   External References:
68      0395 1
69      0396 1
70      0397 1   external routine
71      0398 1       lib$parse: addressing_mode(general),
72      0399 1       phn$answer_cmd,
73      0400 1       phn$break_call,
74      0401 1       phn$conversing,
75      0402 1       phn$dial_cmd,
76      0403 1       phn$directory_cmd,
77      0404 1       phn$exit_cmd,
78      0405 1       phn$facsimile_cmd,
79      0406 1       phn$fresh_screen,
80      0407 1       phn$hangup_cmd,
81      0408 1       phn$help_cmd,
82      0409 1       phn$hold_cmd,
83      0410 1       phn$inform,
84      0411 1       phn$mail_cmd,
85      0412 1       phn$queue_smb,
86      0413 1       phn$reject_cmd,
87      0414 1       phn$show_text,
88      0415 1       phn$transcribe_cmd,
89      0416 1       phn$unhold_cmd;
90      0417 1
91      0418 1   external literal
92      0419 1       lib$_syntaxerr;
93      0420 1
94      0421 1
95      0422 1   Own Variables:
96      0423 1
97      0424 1
```

IN  
V0

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$KBD\_ROUTE - Initial Routing of Input

L 1  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 3  
(3)

```

99 0425 1 %sbttl 'PHN$KBD_ROUTE - Initial Routing of Input'
100 0426 1 ++
101 0427 1 Functional Description:
102 0428 1 This steering message routine is invoked after we collect
103 0429 1 a bit of input from the keyboard. It splits up the input
104 0430 1 into conversation text and command text and routes these
105 0431 1 to the appropriate steering routine for further processing.
106 0432 1
107 0433 1 Formal Parameters:
108 0434 1 input Address of descriptor of input.
109 0435 1
110 0436 1 Implicit Inputs:
111 0437 1 global data
112 0438 1
113 0439 1 Implicit Outputs:
114 0440 1 global data
115 0441 1
116 0442 1 Returned Value:
117 0443 1 none
118 0444 1
119 0445 1 Side Effects:
120 0446 1
121 0447 1 --
122 0448 1
123 0449 1
124 0450 2 global routine phn$kbd_route(input): novalue = begin
125 0451 2
126 0452 2 bind
127 0453 2 input_dsc = .input: descriptor;
128 0454 2
129 0455 2 own
130 0456 2 command_flag: byte initial(false);
131 0457 2
132 0458 2 own
133 0459 2 text_buf: vector[80,byte], ! These variables are OWN due
134 0460 2 buf_i: long; ! up-level referencing.
135 0461 2
136 0462 2 local
137 0463 2 conversing: byte,
138 0464 2 char: byte,
139 0465 2 p: ref pub;
140 0466 2
141 0467 2 bind
142 0468 2 ctrl_w_cmd = describe(%string(%char(ctrl_u,ctrl_w,ret))),
143 0469 2 exit_cmd = describe(%string(%char(ctrl_u),'EXIT',%char(ret))),
144 0470 2 hangup_cmd = describe(%string(%char(ctrl_u),'HANGUP',%char(ret)));
145 0471 2
146 0472 2
147 0473 2 ! The following routine is called when a collection of characters is ready
148 0474 2 ! to be queued to the appropriate steering message routine. If the collection
149 0475 2 ! is non-empty, we queue a steering message. We also clear the buffer for the
150 0476 2 ! next collection.
151 0477 2
152 0478 2 routine queue_collection: novalue = begin
153 0479 2
154 0480 2 local
155 0481 2 collection_dsc: descriptor;
```

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHNSKBD\_ROUTE - Initial Routing of Input

M 1  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 4  
(3)

```

: 156 0482 3
: 157 0483 4 if .buf_i gtru 0 then (
: 158 0484 4     collection_dsc[len] = .buf_i;
: 159 0485 4     collection_dsc[ptr] = text_buf;
: 160 0486 4     phn$queue_smb(if .command_flag then smb__cmd_parse else smb__talk,
: 161 0487 4         collection_dsc);
: 162 0488 4     buf_i = 0;
: 163 0489 3 );
: 164 0490 3 return;
: 165 0491 2 end;
```

.TITLE INPUT INPUT - Analyze and Act on Input  
.IDENT \V04-000\

.PSECT \$SPLITS,NOWRT,NOEXE,2

```

                                OD 17 15 00000 P.AAB: .ASCII <21><23><13>
                                00003
                                00000003 00004 P.AAA: .BLKB 1
                                00000000 00008 .LONG 3
                                OD 54 49 58 45 15 0000C P.AAD: .ADDRESS P.AAB
                                00012 .ASCII <21>\EXIT\<13>
                                00014 P.AAC: .BLKB 2
                                00000006 00018 .LONG 6
                                00000000 0001C P.AAF: .ADDRESS P.AAD
                                OD 50 55 47 4E 41 48 15 0001C P.AAF: .ASCII <21>\HANGUP\<13>
                                00024 P.AAE: .LONG 8
                                00000000 00028 .ADDRESS P.AAF
```

.PSECT \$OWNS,NOEXE,2

```

00 00000 COMMAND_FLAG:
    00001 .BYTE 0
    00004 TEXT_BUF: .BLKB 3
    00054 BUF_i: .BLKB 80
    00054 .BLKB 4
```

CTRL\_W\_CMD= P.AAA  
EXIT\_CMD= P.AAC  
HANGUP\_CMD= P.AAE

```

.EXTRN PHNS_OK, PHNS_ANSWERED
.EXTRN PHNS_BUSYCALL, PHNS_CANCALL
.EXTRN PHNS_CANTREACH, PHNS_CONFCALL
.EXTRN PHNS_DEAD, PHNS_DECNETLINK
.EXTRN PHNS_DIRCAN, PHNS_FACSCAN
.EXTRN PHNS_HELPCAN, PHNS_HUNGUP
.EXTRN PHNS_JUSTRANG, PHNS_LOGGEDOFF
.EXTRN PHNS_REJECTED, PHNS_RING
.EXTRN PHNS_REJECTJUNK
.EXTRN PHNS_SENDINGMAIL
.EXTRN PHNS_BADCMD, PHNS_BADHELP
.EXTRN PHNS_BADMAILCMD
.EXTRN PHNS_BADSMB, PHNS_BADSPEC
.EXTRN PHNS_HELPMISSING
.EXTRN PHNS_IVREDUNANS
.EXTRN PHNS_IVREDUNCALL
```



INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHNSKBD\_ROUTE - Initial Routing of Input

N 1  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 5  
(3)

.EXTRN PHNS\_LINKERROR, PHNS\_NEEDUSER  
.EXTRN PHNS\_NOCALL, PHNS\_NOHOLDS  
.EXTRN PHNS\_NOPTS, PHNS\_NOPRIV  
.EXTRN PHNS\_NOPROC, PHNS\_NOTCONV  
.EXTRN PHNS\_ONLYNODE, PHNS\_PHONEBUSY  
.EXTRN PHNS\_REMOTEERROR  
.EXTRN PHNS\_TARGTERM, PHNS\_UNPLUGGED  
.EXTRN PHNS\_BADTERM, PHNS\_SHAREDMBX  
.EXTRN PHNS\_INPUTTERM, PHNSGQ\_NODE\_NAME  
.EXTRN PHNSGQ\_SWITCH\_HOOK  
.EXTRN PHNSGL\_VIEWPORT\_SIZE  
.EXTRN PHNSGB\_SCROLL, PHNSGQ\_PUBHEAD  
.EXTRN PHNSGB\_FLAGS, LIB\$PARSE  
.EXTRN PHNSANSWER\_CMD, PHNSBREAK\_CALL  
.EXTRN PHNSCONVERSING, PHNSDIAL\_CMD  
.EXTRN PHNSDIRECTORY\_CMD  
.EXTRN PHNSEXIT\_CMD, PHNSFACSIMILE\_CMD  
.EXTRN PHNSFRESH\_SCREEN  
.EXTRN PHNSHANGUP\_CMD, PHNSHELP\_CMD  
.EXTRN PHNSHOLD\_CMD, PHNSINFORM  
.EXTRN PHNSMAIL\_CMD, PHNSQUEUE\_SMB  
.EXTRN PHNSREJECT\_CMD, PHNSSHOW\_TEXT  
.EXTRN PHNSTRANSCRIBE\_CMD  
.EXTRN PHNSUNHOLD\_CMD, LIB\$\_SYNTAXERR

.PSECT \$CODE\$,NOWRT,2

0004 00000 QUEUE_COLLECTION:				
			.WORD Save R2	: 0478
52	0000'	CF 9E 00002	MOVAB BUF_I, R2	:
5E		08 C2 00007	SUBL2 #8, SP	:
50		62 D0 0000A	MOVL BUF_I, R0	: 0483
		1B 13 0000D	BEQL 3\$	:
6E		50 B0 0000F	MOVW R0, COLLECTION_DSC	: 0484
04 AE	B0	A2 9E 00012	MOVAB TEXT_BUF, COLLECTION_DSC+4	: 0485
		5E DD 00017	PUSHL SP	: 0486
04	AC	A2 E9 00019	BLBC COMMAND_FLAG, 1\$	:
		03 DD 0001D	PUSHL #3	:
		02 11 0001F	BRB 2\$	:
		04 DD 00021 1\$:	PUSHL #4	:
0000G CF		02 FB 00023 2\$:	CALLS #2, PHNSQUEUE_SMB	:
		62 D4 00028	CLRL BUF_I	: 0488
		04 0002A 3\$:	RET	: 0491

; Routine Size: 43 bytes, Routine Base: \$CODE\$ + 0000

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$KBD\_ROUTE - Initial Routing of Input

B 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 6  
(4)

```

: 167 0492 2 ! The first thing we have to do is check to see what state the screen is in.
: 168 0493 2 ! If the scroller flag is set, some scrolling-type command (e.g., HELP)
: 169 0494 2 ! is in progress. We need to reset the flag to cancel the command,
: 170 0495 2 ! and refresh the screen.
: 171 0496 2
: 172 0497 2 if .phn$gv_scroller then (
: 173 0498 2     phn$gv_scroller = false;
: 174 0499 2     if ch$find_ch(.input_dsc[.len],.input_dsc[ptr],ctrl_w) eq 0 then
: 175 0500 2         phn$fresh_screen(true);
: 176 0501 2 );
: 177 0502 2
: 178 0503 2 ! Next thing we do is check to see if a message is sitting on the
: 179 0504 2 ! message line. If so, we want to clear it out.
: 180 0505 2
: 181 0506 2 if .phn$gv_message then (
: 182 0507 2     phn$gv_message = false;
: 183 0508 2     phn$inform();
: 184 0509 2 );
: 185 0510 2
: 186 0511 2 ! Another thing we need to check for is whether the user is currently
: 187 0512 2 ! ringing someone's phone. If so, we cancel the call immediately.
: 188 0513 2
: 189 0514 2 p = .phn$gq_pubhead[0];
: 190 0515 2 if .p[pub_v_calling] then
: 191 0516 2     phn$break_call();
: 192 0517 2
: 193 0518 2 ! Finally, we have to see if a facsimile operation is in progress.
: 194 0519 2 ! If so, we cancel it by turning off the flag.
: 195 0520 2
: 196 0521 2 phn$gv_facsimile = false;
```



```

198 0522 2 ! Now we are going to look at the input typed by the user. We will split
199 0523 2 ! it up into conversation text and command text.
200 0524 2
201 0525 2 conversing = phn$conversing();
202 0526 2 buf_i = 0;
203 0527 2
204 0528 2 while dec (input_dsc[len]) geq 0 do (
205 0529 2
206 0530 2     char = ch$rchar_a(input_dsc[ptr]);
207 0531 2
208 0532 2     command_flag = .command_flag or (not .conversing);
209 0533 2     if .command_flag then
210 0534 2
211 0535 2         ! We are in the middle of a command. Process each character:
212 0536 2         !         return      Done with this command, back to talking
213 0537 2         !         switch hook Done with this command, start another.
214 0538 2         !         CTRL/W      Flush current command & refresh screen.
215 0539 2         !         CTRL/Z      Flush current command & force EXIT.
216 0540 2         !         others      Add to current command.
217 0541 2
218 0542 2     select neu .char of set
219 0543 2     [ret]:      (text_buf[.buf_i] = ret;
220 0544 2                 inc(buf_i);
221 0545 2                 queue_collection();
222 0546 2                 command_flag = false;);
223 0547 2
224 0548 2     [ch$rchar(.phn$qq switch_hook[ptr])]:
225 0549 2         (text_buf[.buf_i] = ret;
226 0550 2         inc(buf_i);
227 0551 2         queue_collection(););
228 0552 2
229 0553 2     [ctrl_w]:   (queue_collection();
230 0554 2                 phn$queue_smb(smb_cmd_parse,ctrl_w_cmd);
231 0555 2                 command_flag = false;);
232 0556 2
233 0557 2     [ctrl_z]:   (queue_collection();
234 0558 2                 phn$queue_smb(smb_cmd_parse,exit_cmd);
235 0559 2                 command_flag = false;);
236 0560 2
237 0561 2     [otherwise]: (text_buf[.buf_i] = .char;
238 0562 2                 inc(buf_i););
239 0563 2
240 0564 2 tes

```

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$KBD\_ROUTE - Initial Routing of Input

D 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 8  
(6)

```

: 241 0564 3 else
: 242 0565
: 243 0566
: 244 0567 ! The user is entering some conversation text. Process each
: 245 0568 character as follows:
: 246 0569 switch hook Suspend talking, start a new command.
: 247 0570 CTRL/W Suspend talking, refresh screen.
: 248 0571 CTRL/Z Suspend talking, force HANGUP command.
: 249 0572 others Add to conversation text.
: 250 0573
: 251 0574 select neu .char of set
: 252 0575 [ch$rchar(.phn$gg_switch_hook[ptr]):
: 253 0576 (queue_collection();
: 254 0577 command_flag = true;);
: 255 0578 [ctrl_w]: (queue_collection();
: 256 0579 phn$queue_smb(smb__cmd_parse,ctrl_w_cmd););
: 257 0580
: 258 0581 [ctrl_z]: (queue_collection();
: 259 0582 phn$queue_smb(smb__cmd_parse,hangup_cmd););
: 260 0583
: 261 0584 [otherwise]: (text_buf[.buf_i] = .char;
: 262 0585 inc(buf_i););
: 263 0586 tes;
: 264 0587
: 265 0588 );
: 266 0589
: 267 0590 ! Make sure we queue any final collection we were in the process of building.
: 268 0591
: 269 0592 queue_collection();
: 270 0593 return;
: 271 0594
: 272 0595 1 end;
```

				01FC 00000	.ENTRY PHN\$KBD_ROUTE, Save R2,R3,R4,R5,R6,R7,R8	0450
	58	0000'	CF	9E 00002	MOVAB CTRL_W_CMD, R8	
	57	0000G	CF	9E 00007	MOVAB PHN\$GB_FLAGS, R7	
	56	C6	AF	9E 0000C	MOVAB QUEUE_COLLECTION, R6	
	55	0000'	CF	9E 00010	MOVAB BUF_I, R5	
	53	04	AC	D0 00015	MOVL INPUT, R3	0453
	67		01	E1 00019	BBC #1, PHN\$GB_FLAGS, 2\$	0497
17	67		02	8A 0001D	BICB2 #2, PHN\$GB_FLAGS	0498
04	63		17	3A 00020	LOCC #23, (R3), -24(R3)	0499
			02	12 00025	BNEQ 1\$	
			51	D4 00027	CLRL R1	
			51	D5 00029	TSTL R1	
			07	12 0002B	BNEQ 2\$	
			01	DD 0002D	PUSHL #1	0500
	0000G	CF	01	FB 0002F	CALLS #1, PHN\$FRESH_SCREEN	
	08		67	E9 00034	BLBC PHN\$GB_FLAGS, -3\$	0506
	67		01	8A 00037	BICB2 #1, PHN\$GB_FLAGS	0507
	0000G	CF	00	FB 0003A	CALLS #0, PHN\$INFORM	0508
	50		CF	D0 0003F	MOVL PHN\$GQ_PUBHEAD, P	0514
05	00F0	C0	03	E1 00044	BBC #3, 240(P), 4\$	0515

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHNSKBD\_ROUTE - Initial Routing of Input

E 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1

Page 9  
(6)

0000G	CF		00	FB	0004A	CALLS	#0, PHNSBREAK CALL	0516
	67		08	8A	0004F	BICB2	#8, PHNSGB_FLAGS	0521
0000G	CF		00	FB	00052	CALLS	#0, PHNSCONVERSING	0525
	54		50	90	00057	MOVB	R0, CONVERSING	
			65	D4	0005A	CLRL	BUF_I	0526
	50		63	3C	0005C	MOVZWL	(R3), R0	0528
			50	D7	0005F	DECL	R0	
	63		50	80	00061	MOVW	R0, (R3)	
			50	D5	00064	TSTL	R0	
			03	18	00066	BGEQ	6\$	
			00A5	31	00068	BRW	18\$	
	52	04	B3	90	0006B	MOVB	@4(R3), CHAR	0530
		04	A3	D6	0006F	INCL	4(R3)	
	50		54	9A	00072	MOVZBL	CONVERSING, R0	0532
	51	AC	A5	9A	00075	MOVZBL	COMMAND_FLAG, R1	
	50		51	CA	00079	BICL2	R1, R0	
AC	A5		50	92	0007C	MCOMB	R0, COMMAND_FLAG	
	4F	AC	A5	E9	00080	BLBC	COMMAND_FLAG, 13\$	0533
	0D		52	91	00084	CMPB	CHAR, #13	0543
			10	12	00087	BNEQ	7\$	
	50	BO	A5	9E	00089	MOVAB	TEXT_BUF, R0	
00 B540			0D	90	0008D	MOVB	#13, @BUF_I[R0]	
			65	D6	00092	INCL	BUF_I	0544
	66		00	FB	00094	CALLS	#0, QUEUE_COLLECTION	0545
			35	11	00097	BRB	12\$	0546
0000G	DF		52	91	00099	CMPB	CHAR, @PHNSGQ_SWITCH_HOOK+4	0548
			10	12	0009E	BNEQ	9\$	
	50	BO	A5	9E	000A0	MOVAB	TEXT_BUF, R0	0549
00 B540			0D	90	000A4	MOVB	#13, @BUF_I[R0]	
			65	D6	000A9	INCL	BUF_I	0550
	66		00	FB	000AB	CALLS	#0, QUEUE_COLLECTION	0551
			AC	11	000AE	BRB	5\$	0542
	17		52	91	000B0	CMPB	CHAR, #23	0553
			07	12	000B3	BNEQ	10\$	
	66		00	FB	000B5	CALLS	#0, QUEUE_COLLECTION	
			58	DD	000B8	PUSHL	R8	0554
			0B	11	000BA	BRB	11\$	
	1A		52	91	000BC	CMPB	CHAR, #26	0557
			42	12	000BF	BNEQ	17\$	
	66		00	FB	000C1	CALLS	#0, QUEUE_COLLECTION	
		10	A8	9F	000C4	PUSHAB	EXIT_CMD	0558
			03	DD	000C7	PUSHL	#3	
0000G	CF		02	FB	000C9	CALLS	#2, PHNSQUEUE_SMB	
		AC	A5	94	000CE	CLRB	COMMAND_FLAG	0559
			89	11	000D1	BRB	5\$	0542
0000G	DF		52	91	000D3	CMPB	CHAR, @PHNSGQ_SWITCH_HOOK+4	0574
			09	12	000D8	BNEQ	14\$	
	66		00	FB	000DA	CALLS	#0, QUEUE_COLLECTION	0575
AC	A5		01	90	000DD	MOVB	#1, COMMAND_FLAG	0576
			CB	11	000E1	BRB	8\$	0573
	17		52	91	000E3	CMPB	CHAR, #23	0578
			07	12	000E6	BNEQ	15\$	
	66		00	FB	000E8	CALLS	#0, QUEUE_COLLECTION	
			58	DD	000EB	PUSHL	R8	0579
			0B	11	000ED	BRB	16\$	
	1A		52	91	000EF	CMPB	CHAR, #26	0581
			0F	12	000F2	BNEQ	17\$	



INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$KBD\_ROUTE - Initial Routing of Input

F 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1

Page 10  
(6)

66		00	FB	000F4		CALLS	#0, QUEUE_COLLECTION	:	
	20	A8	9F	000F7		PUSHAB	HANGUP_CMD	:	0582
		03	DD	000FA	16\$:	PUSHL	#3	:	
0000G	CF	02	FB	000FC		CALLS	#2, PHN\$QUEUE_SMB	:	
		AB	11	00101		BRB	8\$	:	0573
	50	A5	9E	00103	17\$:	MOVAB	TEXT_BUF, R0	:	0584
00	B540	52	90	00107		MOVB	CHAR, @BUF_I[R0]	:	
		65	D6	0010C		INCL	BUF_I	:	0585
		9E	11	0010E		BRB	8\$	:	0528
	66	00	FB	00110	18\$:	CALLS	#0, QUEUE_COLLECTION	:	0592
			04	00113		RET		:	0595

; Routine Size: 276 bytes, Routine Base: \$CODE\$ + 002B

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$CMD\_PARSE - Parse a Command

G 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 11  
(7)

```
274 0596 1 %sbttl 'PHN$CMD_PARSE - Parse a Command'
275 0597 1 ++
276 0598 1 Functional Description:
277 0599 1 This steering message routine is invoked when we have part
278 0600 1 or all of a command from PHN$KBD_ROUTE. We collect characters
279 0601 1 until we have a complete command, and then we parse and execute it.
280 0602 1
281 0603 1 Formal Parameters:
282 0604 1 cmd_text Part or all of a command. Never a null string and
283 0605 1 never parts of two commands.
284 0606 1
285 0607 1 Implicit Inputs:
286 0608 1 global data
287 0609 1
288 0610 1 Implicit Outputs:
289 0611 1 global data
290 0612 1
291 0613 1 Returned Value:
292 0614 1 none
293 0615 1
294 0616 1 Side Effects:
295 0617 1
296 0618 1 --
297 0619 1
298 0620 1
299 0621 2 global routine phn$cmd_parse(cmd_text): novalue = begin
300 0622 2
301 0623 2 bind
302 0624 2 text_dsc = .cmd_text: descriptor;
303 0625 2
304 0626 2 own
305 0627 2 command_buf: vector[80,byte],
306 0628 2 buf_i: long initial(0),
307 0629 2
308 0630 2 tparse_block: block[tpa$k_length0,byte]
309 0631 2 initial(tpa$k_count0,
310 0632 2 tpa$m_blanks + tpa$m_abbrev),
311 0633 2 command_proc: long;
312 0634 2
313 0635 2 local
314 0636 2 status: long,
315 0637 2 command_complete: byte,
316 0638 2 char: byte,
317 0639 2 in_quotes: byte, i: long;
```

```
319 0640 2 ! The following is the parsing table used to analyze a command and
320 0641 2 ! determine which command routine to call.
321 0642 2 ! The TRANSCRIBE command has been more or less temporarily removed.
322 0643 2
323 0644 2 $init_state(command_state,command_key);
324 0645 2
325 P 0646 2 $state (,
326 P 0647 2 (tpa$_blank),
327 P 0648 2 (tpa$_lambda)
328 0649 2 );
329 0650 2
330 P 0651 2 $state (,
331 P 0652 2 (tpa$_eos,tpa$_exit),
332 P 0653 2 (ctrl_w, noargs),
333 P 0654 2 ('ANSWER', noargs,,phn$answer_cmd, command_proc),
334 P 0655 2 ('DIAL', args,,phn$dial_cmd, command_proc),
335 P 0656 2 ('DIRECTORY', args,,phn$directory_cmd, command_proc),
336 P 0657 2 ('EXIT', noargs,,phn$exit_cmd, command_proc),
337 P 0658 2 ('FACSIMILE', args,,phn$facsimile_cmd, command_proc),
338 P 0659 2 ('HANGUP', noargs,,phn$hangup_cmd, command_proc),
339 P 0660 2 ('HELP', args,,phn$help_cmd, command_proc),
340 P 0661 2 ('HOLD', noargs,,phn$hold_cmd, command_proc),
341 P 0662 2 ('MAIL', args,,phn$mail_cmd, command_proc),
342 P 0663 2 ('PHONE', args,,phn$dial_cmd, command_proc),
343 P 0664 2 ('REJECT', args,,phn$reject_cmd, command_proc),
344 P 0665 2 ('TRANSCRIBE', args,,phn$transcribe_cmd, command_proc),
345 P 0666 2 ('UNHOLD', noargs,,phn$unhold_cmd, command_proc),
346 P 0667 2 (tpa$_lambda, args,,phn$dial_cmd, command_proc)
347 0668 2 );
348 0669 2
349 P 0670 2 $state (noargs,
350 P 0671 2 (tpa$_blank),
351 P 0672 2 (tpa$_lambda)
352 0673 2 );
353 P 0674 2 $state (,
354 P 0675 2 (tpa$_eos,tpa$_exit)
355 0676 2 );
356 0677 2
357 P 0678 2 $state (args,
358 P 0679 2 (tpa$_blank,tpa$_exit),
359 P 0680 2 (tpa$_lambda,tpa$_exit)
360 0681 2 );
```



```
0682 2 ! We begin by displaying the command fragment on the screen.
0683
0684 phn$show_text(0,text_dsc);
0685
0686 ! We look at each character that has been passed to us. Note that a
0687 similar analysis has to be done in PHN$SHOW_TEXT. There are the
0688 following cases:
0689     normal      Add character to command.
0690     return      End of the command. Parse & execute it.
0691     delete      Delete last character in command.
0692     CTRL/U      Flush entire command and start over.
0693
0694 command_complete = false;
0695 while dec (text_dsc[len]) geq 0 do (
0696     char = ch$rchar_a(text_dsc[ptr]);
0697
0698     select neu .char of set
0699     [%x'20' to %x'7e',
0700      %x'80' to %x'ff']:
0701         (command_buf[.buf_i] = .char;
0702          buf_i = minu(.buf_i+1,79));
0703
0704     [ret]:      command_complete = true;
0705
0706     [delete]:   buf_i = max(.buf_i-1, 0);
0707
0708     [ctrl_u]:   buf_i = 0;
0709     tes;
0710
0711 );
0712
0713 ! If we have a complete command now, we can parse it. All we are interested
0714 in here is the command verb; all arguments will be passed on to the
0715 command processing routine.
0716
0717 if not .command_complete then
0718     return;
0719
0720 ! We need to uppercase the command for comparison purposes. However, any
0721 quoted strings should be left alone.
0722
0723 in_quotes = false;
0724 incr i from 0 to .buf_i-1 do (
0725     if not .in_quotes and
0726         .command_buf[i] gequ 'a' and .command_buf[i] lequ 'z' then
0727         command_buf[i] = .command_buf[i] and %b'11011111';
0728     in_quotes = .in_quotes xor (.command_buf[i] eqlu '"');
0729 );
0730
0731 tparse_block[tpa$l_stringcnt] = .buf_i;
0732 tparse_block[tpa$l_stringptr] = command_buf;
0733 command_proc = 0;
0734 status = lib$tparse(tparse_block,command_state,command_key);
0735
0736 ! If we didn't get a syntax error, we can call the command procedure, passing
0737 it a descriptor of the command arguments (which it can clobber).
0738
```

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$CMD\_PARSE - Parse a Command

J 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 14  
(9)

```
419 0739 2
420 0740 2 if .status eglu lib$ syntaxerr then
421 0741 2     phn$inform(phn$_badcmd)
422 0742 2 else (
423 0743 2     check (.status);
424 0744 2     if .command_proc neqa 0 then
425 0745 2         (.command_proc) (tparse_block[tpa$l_stringcnt]);
426 0746 2 );
427 0747 2
428 0748 2 ! All done. Clear out command line on screen and reset our buffer.
429 0749 2
430 0750 2 phn$show_text(0,describe(%char(ctrl_u)));
431 0751 2 buf_i = 0;
432 0752 2 return;
433 0753 2
434 0754 1 end;
```

```
                                .PSECT _LIB$KEY1$,NOWRT, SHR, PIC,1
                                00000 ;TPASKEYSTO
                                U.9: .BLKB 0
                                52 45 57 53 4E 41 00000 ;TPASKEYST
                                U.11: .ASCII \ANSWER\
                                FF 00006 .BYTE -1
                                00007 ;TPASKEYSTO
                                U.16: .BLKB 0
                                4C 41 49 44 00007 ;TPASKEYST
                                U.18: .ASCII \DIAL\
                                FF 0000B .BYTE -1
                                0000C ;TPASKEYSTO
                                U.24: .BLKB 0
                                59 52 4F 54 43 45 52 49 44 0000C ;TPASKEYST
                                U.26: .ASCII \DIRECTORY\
                                FF 00015 .BYTE -1
                                00016 ;TPASKEYSTO
                                U.31: .BLKB 0
                                54 49 58 45 00016 ;TPASKEYST
                                U.33: .ASCII \EXIT\
                                FF 0001A .BYTE -1
                                0001B ;TPASKEYSTO
                                U.38: .BLKB 0
                                45 4C 49 4D 49 53 43 41 46 0001B ;TPASKEYST
                                U.40: .ASCII \FACSIMILE\
                                FF 00024 .BYTE -1
                                00025 ;TPASKEYSTO
                                U.45: .BLKB 0
                                50 55 47 4E 41 48 00025 ;TPASKEYST
                                U.47: .ASCII \HANGUP\
                                FF 0002B .BYTE -1
                                0002C ;TPASKEYSTO
                                U.52: .BLKB 0
                                50 4C 45 48 0002C ;TPASKEYST
                                U.54: .ASCII \HELP\
                                FF 00030 .BYTE -1
                                00031 ;TPASKEYSTO
```

INPUT - Analyze and Act on Input  
PHNSCMD\_PARSE - Parse a Command

K 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742 P  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1

Page 15  
2:1 (9)

LI  
VO

```

44 4C 4F 48 00031 U.59: .BLKB 0
:TPASKEYST
U.61: .ASCII \HOLD\
FF 00035 .BYTE -1
00036 :TPASKEYSTO
U.66: .BLKB 0
4C 49 41 4D 00036 :TPASKEYST
U.68: .ASCII \MAIL\
FF 0003A .BYTE -1
0003B :TPASKEYSTO
U.73: .BLKB 0
45 4E 4F 48 50 0003B :TPASKEYST
U.75: .ASCII \PHONE\
FF 00040 .BYTE -1
00041 :TPASKEYSTO
U.80: .BLKB 0
54 43 45 4A 45 52 00041 :TPASKEYST
U.82: .ASCII \REJECT\
FF 00047 .BYTE -1
00048 :TPASKEYSTO
U.87: .BLKB 0
44 4C 4F 48 4E 55 00048 :TPASKEYST
U.89: .ASCII \UNHOLD\
FF 0004E .BYTE -1
FF 0004F :TPASKEYFILL
U.98: .BYTE -1

.PSECT _LIB$STATES$,NOWRT, SHR, PIC,1

00000 COMMAND_STATE::
01F2 00000 :TPASTYPE .BLKB 0
05F6 00002 U.2: .WORD 498
:TPASTYPE
11F7 00004 U.3: .WORD 1526
:TPASTYPE
FFFF 00006 U.4: .WORD 4599
:TPASTARGET
1017 00008 U.5: .WORD -1
:TPASTYPE
0000* 0000A U.6: .WORD 4119
:TPASTARGET
7100 0000C U.8: .WORD <<U.7-U.8>-2>
:TPASTYPE
00000000* 0000E U.12: .WORD 28928
:TPASADDR
00000000G 00012 U.13: .LONG <<COMMAND_PROC-U.13>-4>
:TPASMASK
0000* 00016 U.14: .ADDRESS PHN$ANSWER_CMD
:TPASTARGET
7101 00018 U.15: .WORD <<U.7-U.15>-2>
:TPASTYPE
00000000* 0001A U.19: .WORD 28929
:TPASADDR
00000000G 0001E U.20: .LONG <<COMMAND_PROC-U.20>-4>
:TPASMASK
U.21: .ADDRESS PHN$DIAL_CMD

```



INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHNSCMD\_PARSE - Parse a Command

L 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1

Page 16  
(9)

```
0000* 00022 :TPASTARGET
              U.23: .WORD    <<U.22-U.23>-2>
7102 00024 :TPASTYPE
              U.27: .WORD    28930
00000000* 00026 :TPASADDR
              U.28: .LONG    <<COMMAND_PROC-U.28>-4>
00000000G 0002A :TPASMASK
              U.29: .ADDRESS PHNS$DIRECTORY_CMD
0000* 0002E :TPASTARGET
              U.30: .WORD    <<U.22-U.30>-2>
7103 00030 :TPASTYPE
              U.34: .WORD    28931
00000000* 00032 :TPASADDR
              U.35: .LONG    <<COMMAND_PROC-U.35>-4>
00000000G 00036 :TPASMASK
              U.36: .ADDRESS PHNS$EXIT_CMD
0000* 0003A :TPASTARGET
              U.37: .WORD    <<U.7-U.37>-2>
7104 0003C :TPASTYPE
              U.41: .WORD    28932
00000000* 0003E :TPASADDR
              U.42: .LONG    <<COMMAND_PROC-U.42>-4>
00000000G 00042 :TPASMASK
              U.43: .ADDRESS PHNS$FACSIMILE_CMD
0000* 00046 :TPASTARGET
              U.44: .WORD    <<U.22-U.44>-2>
7105 00048 :TPASTYPE
              U.48: .WORD    28933
00000000* 0004A :TPASADDR
              U.49: .LONG    <<COMMAND_PROC-U.49>-4>
00000000G 0004E :TPASMASK
              U.50: .ADDRESS PHNS$HANGUP_CMD
0000* 00052 :TPASTARGET
              U.51: .WORD    <<U.7-U.51>-2>
7106 00054 :TPASTYPE
              U.55: .WORD    28934
00000000* 00056 :TPASADDR
              U.56: .LONG    <<COMMAND_PROC-U.56>-4>
00000000G 0005A :TPASMASK
              U.57: .ADDRESS PHNS$HELP_CMD
0000* 0005E :TPASTARGET
              U.58: .WORD    <<U.22-U.58>-2>
7107 00060 :TPASTYPE
              U.62: .WORD    28935
00000000* 00062 :TPASADDR
              U.63: .LONG    <<COMMAND_PROC-U.63>-4>
00000000G 00066 :TPASMASK
              U.64: .ADDRESS PHNS$HOLD_CMD
0000* 0006A :TPASTARGET
              U.65: .WORD    <<U.7-U.65>-2>
7108 0006C :TPASTYPE
              U.69: .WORD    28936
00000000* 0006E :TPASADDR
              U.70: .LONG    <<COMMAND_PROC-U.70>-4>
00000000G 00072 :TPASMASK
              U.71: .ADDRESS PHNS$MAIL_CMD
0000* 00076 :TPASTARGET
```

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$CMD\_PARSE - Parse a Command

M 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1

Page 17  
(9)

```

      7109 00078 :U.72: .WORD    <<U.22-U.72>-2>      ;
      :TPASTYPE                                     ;
00000000* 0007A :U.76: .WORD    28937                  ;
      :TPASADDR                                     ;
00000000G 0007E :U.77: .LONG    <<COMMAND_PROC-U.77>-4> ;
      :TPASMASK                                     ;
      0000* 00082 :U.78: .ADDRESS PHN$DIAL_CMD          ;
      :TPASTARGET                                     ;
      710A 00084 :U.79: .WORD    <<U.22-U.79>-2>      ;
      :TPASTYPE                                     ;
00000000* 00086 :U.83: .WORD    28938                  ;
      :TPASADDR                                     ;
00000000G 0008A :U.84: .LONG    <<COMMAND_PROC-U.84>-4> ;
      :TPASMASK                                     ;
      0000* 0008E :U.85: .ADDRESS PHN$REJECT_CMD        ;
      :TPASTARGET                                     ;
      710B 00090 :U.86: .WORD    <<U.22-U.86>-2>      ;
      :TPASTYPE                                     ;
00000000* 00092 :U.90: .WORD    28939                  ;
      :TPASADDR                                     ;
00000000G 00096 :U.91: .LONG    <<COMMAND_PROC-U.91>-4> ;
      :TPASMASK                                     ;
      0000* 0009A :U.92: .ADDRESS PHN$UNHOLD_CMD        ;
      :TPASTARGET                                     ;
      75F6 0009C :U.93: .WORD    <<U.7-U.93>-2>        ;
      :TPASTYPE                                     ;
00000000* 0009E :U.94: .WORD    30198                  ;
      :TPASADDR                                     ;
00000000G 000A2 :U.95: .LONG    <<COMMAND_PROC-U.95>-4> ;
      :TPASMASK                                     ;
      0000* 000A6 :U.96: .ADDRESS PHN$DIAL_CMD          ;
      :TPASTARGET                                     ;
      000A8 :U.97: .WORD    <<U.22-U.97>-2>        ;
      :NOARGS                                       ;
      01F2 000A8 :U.7:  .BLKB    0                      ;
      :TPASTYPE                                     ;
      05F6 000AA :U.99: .WORD    498                    ;
      :TPASTYPE                                     ;
      15F7 000AC :U.100: .WORD   1526                   ;
      :TPASTYPE                                     ;
      FFFF 000AE :U.101: .WORD   5623                   ;
      :TPASTARGET                                     ;
      000B0 :U.102: .WORD    -1                        ;
      :ARGS                                         ;
      11F2 000B0 :U.22: .BLKB    0                      ;
      :TPASTYPE                                     ;
      FFFF 000B2 :U.103: .WORD   4594                   ;
      :TPASTARGET                                     ;
      15F6 000B4 :U.104: .WORD    -1                    ;
      :TPASTYPE                                     ;
      FFFF 000B6 :U.105: .WORD   5622                   ;
      :TPASTARGET                                     ;
      :U.106: .WORD    -1                            ;

```

.PSECT \_LIB\$KEY0\$,NOWRT, SHR, PIC,1

00000 COMMAND\_KEY::

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHNSCMD\_PARSE - Parse a Command

N 2  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1

Page 18  
(9)

```

                                .BLKB 0
00000 :TPASKEY0
                                U.1: .BLKB 0
0000* 00000 :TPASKEY
                                U.10: .WORD <U.9-U.1>
0000* 00002 :TPASKEY
                                U.17: .WORD <U.16-U.1>
0000* 00004 :TPASKEY
                                U.25: .WORD <U.24-U.1>
0000* 00006 :TPASKEY
                                U.32: .WORD <U.31-U.1>
0000* 00008 :TPASKEY
                                U.39: .WORD <U.38-U.1>
0000* 0000A :TPASKEY
                                U.46: .WORD <U.45-U.1>
0000* 0000C :TPASKEY
                                U.53: .WORD <U.52-U.1>
0000* 0000E :TPASKEY
                                U.60: .WORD <U.59-U.1>
0000* 00010 :TPASKEY
                                U.67: .WORD <U.66-U.1>
0000* 00012 :TPASKEY
                                U.74: .WORD <U.73-U.1>
0000* 00014 :TPASKEY
                                U.81: .WORD <U.80-U.1>
0000* 00016 :TPASKEY
                                U.88: .WORD <U.87-U.1>

                                .PSECT $SPLITS,NOWRT,NOEXE,2

15 0002C P.AAH: .ASCII <21>
0002D .BLKB 3
00000001 00030 P.AAG: .LONG 1
00000000 00034 .ADDRESS P.AAH

                                .PSECT $OWNS,NOEXE,2

00058 COMMAND_BUF:
                                .BLKB 80
00000000 000A8 BUF_1: .LONG 0
00000003 00000008 000AC TPARSE_BLOCK:
                                .LONG 8, 3
000B4 .BLKB 28
000D0 COMMAND_PROC:
                                .BLKB 4

                                .PSECT $CODE$,NOWRT,2

54 0000' 001C 00000 .ENTRY PHNSCMD_PARSE, Save R2,R3,R4 : 0621
52 04 CF 9E 00002 MOVAB BUF_1, R4 : 0624
AC D0 00007 MOVL CMD_TEXT, R2 : 0684
52 DD 0000B PUSHL R2
7E D4 0000D CLRL -(SP)
0000G CF 02 FB 0000F CALLS #2, PHNS$SHOW TEXT
53 94 00014 CLRB COMMAND_COMPLETE : 0694
```



INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHNSCMD\_PARSE - Parse a Command

B 3  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1

Page 19  
(9)

50		62	3C	00016	1\$:	MOVZWL	(R2), R0	0695
		50	D7	00019		DECL	R0	
62		50	B0	0001B		MOVW	R0, (R2)	
		50	D5	0001E		TSTL	R0	
		5A	19	00020		BLSS	8\$	
51	04	B2	90	00022		MOVB	24(R2), CHAR	0697
	04	A2	D6	00026		INCL	4(R2)	
20		51	91	00029		CMPB	CHAR, #32	0700
		06	1F	0002C		BLSSU	2\$	
7E	8F	51	91	0002E		CMPB	CHAR, #126	
		06	1B	00032		BLEQU	3\$	
80	8F	51	91	00034	2\$:	CMPB	CHAR, #128	
		1C	1F	00038		BLSSU	4\$	
50		A4	9E	0003A	3\$:	MOVAB	COMMAND_BUF, R0	0702
00	B440	51	90	0003E		MOVB	CHAR, @BUF_I[R0]	
50		01	C1	00043		ADDL3	#1, BUF_I, R0	0703
0000004F	8F	50	D1	00047		CMPL	R0, #79	
		1E	1B	0004E		BLEQU	6\$	
50		8F	9A	00050		MOVZBL	#79, R0	
		18	11	00054		BRB	6\$	
0D		51	91	00056	4\$:	CMPB	CHAR, #13	0705
		05	12	00059		BNEQ	5\$	
53		01	90	0005B		MOVB	#1, COMMAND_COMPLETE	
		B6	11	0005E		BRB	1\$	
7F	8F	51	91	00060	5\$:	CMPB	CHAR, #127	0707
		0D	12	00064		BNEQ	7\$	
50		01	C3	00066		SUBL3	#1, BUF_I, R0	
		02	18	0006A		BGEQ	6\$	
		50	D4	0006C		CLRL	R0	
64		50	D0	0006E	6\$:	MOVL	R0, BUF_I	
		A3	11	00071		BRB	1\$	
15		51	91	00073	7\$:	CMPB	CHAR, #21	0709
		9E	12	00076		BNEQ	1\$	
		64	D4	00078		CLRL	BUF_I	
		9A	11	0007A		BRB	1\$	0695
01		53	E8	0007C	8\$:	BLBS	COMMAND_COMPLETE, 9\$	0718
			04	0007F		RET		
		52	94	00080	9\$:	CLRB	IN_QUOTES	0724
50		01	CE	00082		MNEGL	#1, I	0725
		27	11	00085		BRB	13\$	
16		52	E8	00087	10\$:	BLBS	IN_QUOTES, 11\$	0726
61	8F	B0	A440	91	0008A	CMPB	COMMAND_BUF[I], #97	0727
		0E	1F	00090		BLSSU	11\$	
7A	8F	B0	A440	91	00092	CMPB	COMMAND_BUF[I], #122	
		06	1A	00098		BGTRU	11\$	
B0	A440	20	8F	8A	0009A	BICB2	#-224, COMMAND_BUF[I]	0728
		51	D4	000A0	11\$:	CLRL	R1	0729
22		B0	A440	91	000A2	CMPB	COMMAND_BUF[I], #34	
		02	12	000A7		BNEQ	12\$	
		51	D6	000A9		INCL	R1	
52		51	2C	000AB	12\$:	XORB2	R1, IN_QUOTES	
D5		64	F2	000AE	13\$:	AOBLSS	BUF_I, I, 10\$	0725
		64	D0	000B2		MOVL	BUF_I, TPARSE_BLOCK+8	0732
OC	A4	B0	A4	9E	000B6	MOVAB	COMMAND_BUF, TPARSE_BLOCK+12	0733
10	A4	28	A4	D4	000BB	CLRL	COMMAND_PROC	0734
		0000	CF	9F	000BE	PUSHAB	COMMAND_KEY	0735
		0000	CF	9F	000C2	PUSHAB	COMMAND_STATE	

LI  
VO

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$CMD\_PARSE - Parse a Command

C 3  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1

Page 20  
(9)

00000000G	00	04	A4	9F	000C6	PUSHAB	TPARSE_BLOCK	:	
	52		03	FB	000C9	CALLS	#3, LIB\$TPARSE	:	
00000000G	8F		50	D0	000D0	MOVL	R0, STATUS	:	
			52	D1	000D3	CMPL	STATUS, #LIB\$_SYNTAXERR	:	0740
			0D	12	000DA	BNEQ	14\$	:	
		00000000G	8F	DD	000DC	PUSHL	#PHN\$_BADCMD	:	0741
0000G	CF		01	FB	000E2	CALLS	#1, PHN\$INFORM	:	
			18	11	000E7	BRB	16\$	:	
	09		52	E8	000E9	BLBS	STATUS, 15\$	:	0743
			52	DD	000EC	PUSHL	STATUS	:	
00000000G	00		01	FB	000EE	CALLS	#1, LIB\$SIGNAL	:	
	50	28	A4	D0	000F5	MOVL	COMMAND_PROC, R0	:	0744
			06	13	000F9	BEQL	16\$	:	
		0C	A4	9F	000FB	PUSHAB	TPARSE_BLOCK+8	:	0745
	60		01	FB	000FE	CALLS	#1, (R0)	:	
		0000'	CF	9F	00101	PUSHAB	P.AAG	:	0750
			7E	D4	00105	CLRL	-(SP)	:	
0000G	CF		02	FB	00107	CALLS	#2, PHN\$SHOW_TEXT	:	0751
			64	D4	0010C	CLRL	BUF_I	:	0754
			04	00	0010E	RET		:	

: Routine Size: 271 bytes, Routine Base: \$CODE\$ + 013F

: 435 0755 1  
: 436 0756 0 end eludom

.EXTRN LIB\$SIGNAL

#### PSECT SUMMARY

Name	Bytes	Attributes					
\$OWNS	212	NOVEC, WRT, RD	, NOEXE, NOSHR,	LCL, REL,	CON, NOPIC,	ALIGN(2)	
\$PLITS	56	NOVEC, NOWRT, RD	, NOEXE, NOSHR,	LCL, REL,	CON, NOPIC,	ALIGN(2)	
\$CODE\$	590	NOVEC, NOWRT, RD	, EXE, NOSHR,	LCL, REL,	CON, NOPIC,	ALIGN(2)	
_LIB\$KEY0\$	24	NOVEC, NOWRT, RD	, EXE, SHR,	LCL, REL,	CON, PIC,	ALIGN(1)	
_LIB\$STATES	184	NOVEC, NOWRT, RD	, EXE, SHR,	LCL, REL,	CON, PIC,	ALIGN(1)	
_LIB\$KEY1\$	80	NOVEC, NOWRT, RD	, EXE, SHR,	LCL, REL,	CON, PIC,	ALIGN(1)	

#### Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:00.7
\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	24	57	14	00:00.1

INPUT  
V04-000

INPUT - Analyze and Act on Input  
PHN\$CMD\_PARSE - Parse a Command

D 3  
16-Sep-1984 02:10:30  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[PHONE.SRC]INPUT.B32;1  
Page 21  
(9)

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:INPUT/OBJ=OBJ\$:INPUT MSRC\$:INPUT/UPDATE=(ENH\$:INPUT)

: Size: 590 code + 556 data bytes  
: Run Time: 00:16.8  
: Elapsed Time: 01:03.8  
: Lines/CPU Min: 2703  
: Lexemes/CPU-Min: 87343  
: Memory Used: 200 pages  
: Compilation Complete



0305 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY